APPENDIX A

3. PDX Messages

3.1. Required Messages

Since the PDX Service Adapter is the HUB of the Integration architecture, it does not adhere to the normal Required or Optional Message's policy that the rest of the PDX adapters use, however, it must implement the same adapter interfaces. The PDX Adapter accepts incoming messages that are not part of the individual Application's Service Adapter. All messages described below are considered required for the PDX adapter. No other adapter needs to implement these messages.

3.1.1. checkLinks_RQ

3.1.1.1. Data/Business Rules

This message will check to see if a link or links exist for a person. The ID of the Person is retrieved from the Messages Source Section. The Source section also contains the Source Application's information(Service Adapter Name and Data Source Description). Using this information the PDX Adapter will query the Links table for each Destination supplied in the Message to see if there is a link for a person between the Source Application and the Destination Applicaiton.

3.1.1.2. Parameters

Name	Description
PersonID	This is the Unique for a Person in the Native Application (Source).

3.1.1.3. Results

Name	Value(s)
LinkExists	True or False
DestinationID	The Native Application's Identifier for the Person if a link was found.
PartyID	The Person's PartyID, or Unique identifier as defined by PDX.

3.1.2. checkLinksForApps_RQ

3.1.2.1. Data/Business Rules

This Message will return, for a particular application, wheter or not links exist for that application. Note that this message does not care about one particular person, rather it just checks to see if any links exist for this application

3.1.2.2. Parameters

Name	Description
UserAppID	This is the Unique identifier as defined in PDX for a PDX Enabled Application. Note: This message can process one or more UserAppIDs.

3.1.2.3. Results

Name	Value(s)
LinksExist	True or False.

3.1.3. getLinks_RQ

3.1.3.1. Data/Business Rules

This message will return the Link information for a particular Person if found. It uses the information supplied in the incoming Message's Source Section (PersonID, ServiceAdapterName, Data Source Description). It first obtains the UserAppID of the Source Application using the information mentioned above. Once the UserAppID of the Source has been obtained, it then loops through all of the Destination's in the incoming message.

This message will add a PDXLinksSection to the Payload of the message. Please see Results below for information in the PDXLinksSection.

3.1.3.2. Parameters

Name	Description
PersonID	This is the Unique for a Person in the Native Application (Source).

3.1.3.3. Results

If a link was found for a person, based on the Source Applicaiton and the Destination application the following fields will be valued. If NO Link was found, the following fields will be created, but not valued.

Name	Description
ActionPerformed	This is a placeholder for the Destination Adapter to value. The Destination Adapter will value this after a SyncPerson Message with either Added or Updated.
ObjectType	Valued with PERSON

Name	Description
SourceID	The PersonID of the Person in the Source Application.
DestinationID	The PersonID of the Person in the Destination Application.
PartyID	The Unique Identifier for a Person as defined by PDX.
SourceAppID	The UserAppID of the Source Application.
SignatureDate	This is a placeholder for the Destination Service Adapter to value. The value of this will be the Date/Time that the last SyncPerson took place.

3.1.4. updateLinks_RQ

3.1.4.1. Data/Business Rules

This message is called after a SyncPerson has been performed. It used the PDXLinkSection mentioned in the getLinks_RQ Message. After a SyncPerson has been performed, the Destination Service Adapter values fields in this PDXLinksSection. The PDX Adapter then uses this information Update the Links Table.

Two actions can be taken depending on the Value of the ActionPerfomed property. If the Action Performed is Added, then a new Link will be added for the Person based on the Source Application and Destination Application.

If the Result is Added, then this message will also check to see if a Party (Unique Identifier as defined by PDX) exists for this Person. If one does not exist, it will create a new one. If one does exist, it will used this PartyID during the Creation of the Link.

If the Result is Updated, the existing Link is simply updated with the values in the PDXLinksSection.

The LastUpdated Date in the Links table will also be updated during this process. This value is also retrieved from the PDXLinksSection.

3.1.4.2. Parameters

None

3.1.4.3. Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.5. addLinks_RQ

3.1.5.1. Data/Business rules

This message performs the same processing as updateLinks_RQ. Please see updateLinks_RQ.

3.1.5.2. Parameters

Please see updateLinks_RQ.

3.1.5.3. Results

Please see updateLinks_RQ.

3.1.6. clearLinks_RQ

3.1.6.1. Data/Business rules

This message will clear all of the existing Links for an Application. The message uses information defined in the Destination Section of the incoming message to obtain the UserAppID for an Application. Once the UserAppID has been obtained, all of the Links in the Links table will be removed for that UserAppID.

Additionally, a query will be executed against the Links table to remove all Links for a Person where there is only one Link. It will also delete every Person in the Party table where there is only one Link. This means that if a person only has one Link in the Links table, that Link will be deleted, because a single Link is not useable.

3.1.6.2. Parameters

Name	Description
Uses the Destination Section of the Message.	PersonID, ServiceAdapterName and DataSourceDescription.

3.1.6.3. Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.7. clearLinksByDate_RQ

3.1.7.1. Data/Business rules

This message is very similar to the clearLinks_RQ message. The only difference is that it will clearLinks based on a Date rather that clearing all of the Links for a particular application.

The Links that will be delete are all of those whose LastUpdated Date in the Links table is greater than the date supplied. The date supplied is the signature date of a particular application.

Also, like the clearLinks_RQ message, all single links that remain, and associated Parties, will also be deleted as a result of this message.

3.1.7.2. Parameters

Name	Description
SignatureDate	This is the signature date of the Destination Application.
Destination Section	This message also uses the Destination section to obtain the UserAppID of the Destination Application.

3.1.7.3. Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.8. getDashboards_RQ

3.1.8.1. Data/Business rules

This message will return the Dashboard and all applications currently defined to the dashboard for a particular user.

The Adapter will attempt to retrieve the dashboard for the user id specified in the IONSID property of the incoming message. If no dashboard is found for this User, a new one will be created using the default user's dashboard. The default user is specified in the database with an IONSID = 'X000000'. This data comes pre-populated with PDX.

3.1.8.2. Parameters

Name	Description
Type	CURRENT - Get the Dashboard for a user where the MostRecentlyUsed column is equal to 'Y' USER - Get the Dashboard for a user where the IONSID is equal to the incoming Message's IONSID propety. DEFAULT - Get the Dashboard Where the IONSID = 'X000000'
IONSID	Although not a Parameter, it uses this value to retrieve the dashboard for the user. This is specified in the IONSID property of the message.

3.1.8.3. Results

The Results of the message is a Payload with a Single Dashboard and one or more Application objects.

Below Defines the Dashboard Properties.

Name	Description
DashboardID	The Unique identifier for the Dashboard.
DashboardName	The Name that the user specifies for this Dashboard.
LastPartyID	The PersonID of the Client last used within this dashboard.
MostRecentlyUsed	Indicator that specifies that this was the Dashboard the user last used.
UsageCount	Not Used.
WarnBeforeSync	Indicator specifying whether or not to display a warning message to the user prior to performing a Sync. *For Release 1.0 this is always 'Y'
LastAppIndex	Used in combination with the LastPartyID. This defines the application the LastPartyID belongs to.
Applications	Collection. See Below

Below Defines the Application Properties.

Name	Description
ApplicationID	Unique Identifier for the Top Level Application.
ServiceAdapterName	The Name of the ServiceAdapter to use when making PDX requests.
AppDescription	The Top Level Description for the Application.
AppLaunchString	The Command to use to Start or Launch the Application.
AppParameterString	The Parameters to use, if any, to use when Starting or Launching the Application.
*MembershipLevel	Defines the PDX capabilities for this application.
ApplicationOrder	The order in which to display this application within the Dashboard.

Name	Description
SourceDescription	The name of the Data Source to be used in this application. Most of the time, for a database, it is the DSN.
UserID	The UserName to use for this application and SourceDescription.
DisplayName	The Name of the Application, defined by the user. Used when displaying the application within the Dashboard.
CanLaunch	Indicator that defines whether an application can be launched or Started from the Dashboard.
UserAppID	The unique identifier for a user defined Application.
Validated	Indicator that determines whether or not this application's settings or properties have been validated by the User. If the Application has not been validated, it cannot be used within the Dashboard.
PDXSignature	Constant, always set to VALID.

*See Section 4.1 for further Information.

3.1.9. getPDXApplications_RQ

3.1.9.1. Data/Business rules

This message will return all Applications that are PDX Enabled. This is determined if the Application's Membership Level is greater than zero.

*Please See Section 4.1 Definitions of Membership Level.

Note that this message will NOT return the User Defined Applications, (UserApps) rather it will return the Top Level definition for an application.

3.1.9.2. Parameters

None.

3.1.9.3. Results

The Results of the message is a Payload with a Collection of Application Objects.

Name	Description
ApplicationID	Unique Identifier for the Top Level Application

Name	Description
	Application.
ServiceAdapterName	The Name of the ServiceAdapter to use when making PDX requests.
AppDescription	The Top Level Description for the Application.
AppLaunchString	The Command to use to Start or Launch the Application.
AppParameterString	The Parameters to use, if any, to use when Starting or Launching the Application.
*MembershipLevel	Defines the PDX capabilities for this application.

*See Below for further Definition.

3.1.10. addApplications_RQ

3.1.10.1.Data/Business rules

This Message will add an Application to a Dashboard using properties given by the User. Once the User, via the Dashboard, selects an application, they must make changes, such as UserName, Password, DataSource selections. These selections are then saved to the UserApps Table within the PDX Database. The data about the Applicaiton to be added comes in the Payload Section of the incoming message.

The Adapter will first query the UserApps table to determine if this application and its settings are already associated with the specified dashboard. If an entry is found in the UserApps, the Adapter will make the association with the Dashboard and the Application by inserting a row into the DashboardApps table.

In order to determine if the application already exists in the UserApps table, it queries the UserApps Table with the following Criteria:

IONSID = Message's IONSID property.
SourceDescription = Application to be added's SourceDescription Property.
AppLaunchString = Application to be added's AppLaunchString Property.
AppParameterString = Application to be added's AppParameterString Property.

3.1.10.2.Parameters

Name	Description

Name	Description
DashboardID	The ID of the Dashboard in which to associate the newly added application with.
Application Object	Although not a parameter, this information come in the Payload seciton of the Message. See below for all of the Properties sent in this object.

Application Object Properties.

Name	Description
ApplicationID	Unique Identifier for the Top Level Application.
*ApplicationOrder	The order in which to display this application within the Dashboard.
*SourceDescription	The name of the Data Source to be used in this application. Most of the time, for a database, it is the DSN.
*UserID	The UserName to use for this application and SourceDescription.
*DisplayName	The Name of the Application, defined by the user. Used when displaying the application within the Dashboard.
CanLaunch	Indicator that defines whether an application can be launched or Started from the Dashboard.
*AppLaunchString	The Command to use to Start or Launch the Application.
*AppParameterString	The Parameters to use, if any, to use when Starting or Launching the Application.
ServiceAdapterName	The Name of the ServiceAdapter to use when making PDX requests.
Validated	Indicates whether or not the Application was Validated by the User.

*Indicates all values that can be modified by the User.

3.1.10.3.Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.11. updateApplications_RQ

3.1.11.1.Data/Business rules

This Message will is very similar to the AddApplications, only this message will update an existing Application in the UserApps Table. The Adapter does perform some more initial processing when updating an application. This processing is described below.

Like the addApplications Message, the Adapter will attempt to fine an existing application with the same properties in the UserApps Table. The query will check for IONSID, SourceDescription, AppLaunchString and AppParameterString.

If no match was found the Adapter will simply update the UserApps table with the supplied information using the supplied UserAppID as the Key.

If an Applicaiton was found in the query on the UserApps table, two conditions are checked.

1. If the UserAppID found is different than the one supplied.
2. If the UserAppID found is the same as the one supplied.

If Condition #1 is true, then the Adapter will update the row in the UserApps table where the UserAppID is equal to the one found in the database, not the one supplied in the message.

If Condition #2 is true, then the Adapter will update the row in the UserApps table where the UserAppID is equal to the one supplied in the message.

The Application order column that is found in the DashboardApps table will also be updated with value supplied in the message.

3.1.11.2.Parameters

Name	Description
DashboardID	The ID of the Dashboard in which to associate the newly added application with.
Application Object	Although not a parameter, this information come in the Payload seciton of the Message. See below for all of the Properties sent in this object.

Application Object Properties.

Name	Description
ApplicationID	Unique Identifier for the Top Level Application.
*ApplicationOrder	The order in which to display this application within the Dashboard.
*SourceDescription	The name of the Data Source to be used in this application. Most of the time, for a database, it is the DSN.
*UserID	The UserName to use for this application and SourceDescription.
*DisplayName	The Name of the Application, defined by the user. Used when displaying the application within the Dashboard.
CanLaunch	Indicator that defines whether an application can be launched or Started from the Dashboard.
*AppLaunchString	The Command to use to Start or Launch the Application.
*AppParameterString	The Parameters to use, if any, to use when Starting or Launching the Application.
ServiceAdapterName	The Name of the ServiceAdapter to use when making PDX requests.
Validated	Indicates whether or not the Application was Validated by the User.

*Indicates all values that can be modified by the User.

3.1.11.3.Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.12. removeApplications_RQ

3.1.12.1.Data/Business rules

This Message will remove an application from a Dashboard. The entry in the UserApps Table for the Application to be removed will not be deleted, however the association between the Dashboard and the UserApp will be deleted. We do not want to delete the information from the UserApps table because there may be Links in the Database for this Application. If

the User decided to re-add the Application to the Dashboard at a later date, the Links will remain.

3.1.12.2.Parameters

Name	Description
DashboardID	The ID of the Dashboard in which to associate the newly added application with.
Application Object	Although not a parameter, this information come in the Payload seciton of the Message. See below for all of the Properties sent in this object.

Application Object Properties.

Name	Description
UserAppID	This is the Unique identifier as defined in PDX for a PDX Enabled Application. Note: This message can process one or more UserAppIDs.

3.1.12.3.Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.13. addDashboard_RQ

3.1.13.1.Data/Business rules

This Message will add a row to the Dashboard Table using information supplied in the Payload to value the columns.

3.1.13.2.Parameters

Although not parameters, the following information is used when creating a dashboard. This information comes in the Payload of the Message.

Name	Description
IONSID	The UserID of the User who is creating this Dashboard.
DashboardName	The Name to be given to the Dashboard. Supplied by the User.

3.1.13.3.Results

The DashboardID is Returned in the Payload on Successful completion of the message.

3.1.14. updateDashboard_RQ

3.1.14.1.Data/Business rules

This Message will update properties of an existing Dashboard based on the DashboardID supplied in the Parameters Section. Properties of the Dashboard are found in the Payload section of the Incoming Message.

3.1.14.2.Parameters

Name	Description
DashboardID	The ID of the Dashboard in which to associate the newly added application with.
Dashboard Object	Although not a parameter, this information come in the Payload seciton of the Message. See below for all of the Properties sent in this object.

Dashboard Object Properties.

Name	Description
*DashboardName	The Name that the user specifies for this Dashboard.
LastPartyID	The PersonID of the Client last used within this dashboard.
MostRecentlyUsed	Indicator that specifies that this was the Dashboard the user last used.
UsageCount	Not Used.
WarnBeforeSync	Indicator specifying whether or not to display a warning message to the user prior to performing a Sync. *For Release 1.0 this is always 'Y'
LastAppIndex	Used in combination with the LastPartyID. This defines the application the LastPartyID belongs to.

*Indicates all values that can be modified directly by the User.

3.1.14.3.Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.15. removeDashboard_RQ

3.1.15.1.Data/Business rules

This Message will delete a Dashboard from the PDXDatabase. All rows in the Dashboard Apps table will be deleted based on the DashboardID supplied in the Parameters section. Next, the row in the Dashboards Table will be deleted. All rows in the UserApps table will remain, as they may have associated Links in them. If the user chooses to add those applications back to a new dashboard, those Links will remain.

3.1.15.2.Parameters

Name	Description
DashboardID	The ID of the Dashboard in which to associate the newly added application with.

3.1.15.3.Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.16. setDashboardAppOrder_RQ

3.1.16.1.Data/Business rules

This Message will set an application's display order within a given dashboard.

3.1.16.2.Parameters

Name	Description
DashboardID	The ID of the Dashboard in which to associate the newly added application with.
IONSID	Not a Parameter, comes from the IONSID Property of the Message Object.
Application Object	Although not a parameter, this information come in the Payload seciton of the Message. See below for all of the Properties sent in this object.

Application Object Properties.

Name	Description
ApplicationOrder	The order in which to display this application within the Dashboard.
UserAppID	This is the Unique identifier as defined in PDX for a PDX Enabled Application. Note: This message can process one or more UserAppIDs.

3.1.16.3.Results

If successful, it will return a successful completion message. Otherwise an appropriate Error will be returned.

3.1.17. restoreDefaultDashboardApps_RQ

3.1.17.1.Data/Business rules

This Message will restore the current Dashboard to it's default settings, including the application's in the Dashboard. All customizations by the user to the Dashboard will be lost.

The first action to take place is to delete all of the entries in the Dashboard Apps table using the DashboardID supplied as a parameter.

The next action to take place is to initialize columns in the existing dashboard to their default values. These are: WarnBeforeSync = 'Y', LastPartyID = 0, lastAppIndex = -1.

Next, a query is executed that gets all of the UserApps for the Default User X000000. For every UserApp in for the Default User a query is executed on the UserApps table using the IONSID specified in the message and the UserApp's Application id as search criteria. If no match was found, a new row is inserted. However, if a match was found, it will update that UserApp's information with the information found in the Default User's UserApp. Once the addition or update takes place on the UserApps table, a new row is inserted into the Dashboard Apps table to make the association between the Dashboard and the UserApp.

3.1.17.2.Parameters

Name	Description
DashboardID	The ID of the Dashboard in which to associate the newly added application with.
IONSID	Not a Parameter, comes from the IONSID Property of the Message Object.

3.1.17.3.Results

Returns Success or the appropriate Error Message.

3.2. Optional Messages

There are no Optional Messages for the PDX Service Adapter.

4. Special Processing

4.1. Membership Level Definitions

Below is a table defining the Membership Level for Applications within PDX.

Membership Level Value	Definition
1	Application has Add Only Capabilities.
2	Application has Search Capabily.
4	Application has Get capability, meaning it can retrieve information about a Client based on an ID.
8	Application has Synchronize Capability, both in an Add mode and an Update Mode.
15	Application has all of the above mentioned capabilities, Add, Search, Get and Synchronize.
0	Non PDX Application

APPENDIX B

3. PDX Messages

3.1. Required Messages

Required Messages are those that must be implemented within a Service Adapter. These messages are mostly related to configuration and capabilities of the individual adapter.

3.1.1. getSettings_RQ

The get Settings message requests information from the Service Adapter so that the core PDX functionality can properly allow a user to configure the service adapter.

3.1.1.1. Parameters

None

3.1.1.2. Results

Name	Value(s)
Has Multiple Data Sources	Yes or No
Has User Names	Yes or No
Has Passwords	Yes or No

3.1.2. validateSettings_RQ

The validate settings message requests that a Service Adapter validate all of the necessary information that will allow it to participate in PDX.

3.1.2.1. Parameters

Name	Description
EXE Name	The Name and location of the executable used to launch the application.
User Name	The User Name for the application.
Password	The Password associated with the user name.
Source Description	The DSN or file location of the data store.

3.1.2.2. Results

If the settings are all valid then it returns the following:

Name	Value(s)
CanLaunch	Yes or No - The application can be launched
Can Add	Yes or No - The Service Adapter can add new persons.
Can Search	Yes or No - The Service Adapter has search capability.
Can Get	Yes or No - The Service Adapter can return person information.
Can Sync	Yes or No - The Service Adapter has the capability to synchronize information.

If one or more settings are not valid, an appropriate response will be given, for example.

Error Number: **PDX1008**
Error Description: **Invalid UserName/Password Combination.**

Error Number: **PDX1010**
Error Description: **The EXE Name / Target is invalid for this application.**

Error Number: **PDX10??**
Error Description: **The Data Source Selected is not a ??? Data Source.**

3.1.3. getPDXSignature_RQ

The getPDXSignature message will return a value that indicates whether or not a PDX Signature exits for the datasource. A PDX Signature identifies to PDX that an application's data store has not been corrupted and subsequently been rebuilt. If a datastore has been rebuilt then the unique id's for clients within that datastore are more than likely no longer valid. If a Signature row does not exist, PDX will have to clear all existing links for that particular datastore.

When attempting to retrieve the signature row, it should be retrieved for the user id within the message. This user ID is the Windows 2000 logon ID. In most cases this will be the IONSID.

3.1.3.1. Parameters

None.

3.1.3.2. Results

If the PDX Signature is found, nothing is returned, however, the message's status will indicate a successful completion.

If the PDX Signature is not found, an error and its code are placed within the message.

Error Number: **PDX1005**
Error Description: **PDX Signature Not Found.**

3.1.4. addPDXSignature_RQ

This message will add a signature row or record to the Service Adapter's datastore. The signature row will be added for the user that is sending the message. There should be a signature row for each user, not just one PDX Signature row.

3.1.4.1. Parameters

None

3.1.4.2. Results

The Service Adapter should first check to ensure that a signature row does not exist for the current user. If one already exists the following error will be placed within the message

Error Number: **PDX1006**
Error Description: **PDX Signature Already Exists.**

If one does not already exist, the message will just return successuful.

3.1.5. getDataSources_RQ

This message will return all available data sources / file location's for a the Service Adapter's application.

3.1.5.1. Parameters

None

3.1.5.2. Results

Name	Value(s)
SourceType	ODBC or FILE
Name	The Name of the Data Source or File.

There will only be one occurrence for SourceType, but there can be one or more occurrences of Name.

3.1.6. getUserNames_RQ

This message will return all available User Names for a given data source.

3.1.6.1. Parameters

Name	Description
SourceDescription	This is the Data Source, either DSN or File Location that you want to get a list of User Names for.

3.1.6.2. Results

Name	Value(s)
Name	The User Names.

There can be multiple instances of Name.

3.2. Optional Messages

Optional Messages are those that are not required by the adapter. For instance, if we send a message to an adapter that doesn't store a particular group of information, such as products, the service adapter does not have to implement this message. However, it must inform the integration engine that the message just received is not supported.

3.2.1. searchPerson_RQ

This message will perform a search based on one or more parameters given. The method of searching is up to the Service Adapter. For example, when searching for People with a First Name of 'John', the Adapter may choose to search where the first name equals 'John', or, the Adapter may choose to search where the first name begins with 'John'. In the second example it would return all names that begin with John (ie. John, Johnathan, Johnson....)

3.2.1.1. Parameters

Name	Description
FirstName	The Person's First Name.
LastName	The Person's Last Name.
BirthDate	The Person's Date of Birth.
GovernmentID	The Person's Government ID or Social Security Number.
PostalCode	The Zip code of the Person's primary address.

PersonID	This is the Unique Identifier for an individual within an application.

3.2.1.2. Results

Name	Description
PersonID	This is the Unique Identifier for an individual within an application.
FirstName	The Person's First Name.
MiddleName	The Person's Middle Name.
LastName	The Person's Last Name.
NameSuffix	The Person's Name Suffix. Ie(Jr., Sr.)
GovernmentId	The Person's Government ID or Social Security Number.
BirthDate	The Person's Date of Birth.
The Following fields are the Person's address information. It should be the Person's primary address.	
Line1	
Line2	
Line3	
City	
State	
PostalCode	

3.2.2. getPerson_RQ

This message will return all of the demographic data associated with a person as defined by PDX.

3.2.2.1. Parameters

Name	Description
PersonID	This is the Unique Identifier for an individual within an application.

3.2.2.2. Results

Please refer to the PDX Business Requirements document section 5.3 data requirements for this information.

3.2.3. syncPerson_RQ

This message will synchronize data from one application's data store (source) to one or more (destination) data stores. This message actually performs multiple functions. It must first get the data from the source, then check to see if there are already links associated with this person for the source and destination(s). Then it will perform the synchronization. Once the synchronization is complete, it will then add or update the links for the source and destination(s) if necessary.

3.2.3.1. Parameters

Name	Description
PersonID	This is the Unique Identifier for an individual within an application.

3.2.3.2. Results

The result of this message will simply be, successful or not. If an error was encountered during the processing, this error will be included in the message along with information about the error itself, such as, number description etc...

4. PDX Message Structure.

In order to accomplish integration with multiple applications that are built in many different ways and by different groups both within Prudential and by eternal vendors, these applications must communicate using a common language. This common language is a must for this strategy to be successful.

The PDX Message is the foundation of the common language and must be understood by all Service adapters who are participating in PDX. The structure of the message remains the same regardless of the type of message being sent.

The Message itself is broken into two main sections, the Envelope and the Body. Above these main sections is what we call the root of the message.

4.1. Message Root

The Message Root simply contains the IONS ID and Request Type.

Property	Description
IONSID	This is the Windows 2000 log on id, or the id in which the user logged onto Windows.
RequestType	This is the actual request being made via the message. (Ex. GetPerson_RQ, SearchPerson_RQ)

4.2. Message Envelope

The Envelope section of the message contains routing information for the message. It lets the integration engine know which Service Adapter(s) to send the message to. So, it basically contains Source information (ie. where to get the data from) and Destination information (where to send the request too). There will be cases where Source information is not needed, so the initiator of the message will not need to supply Source information.

4.2.1. Source

The Source Service Adapter's information. Only one occurrence can exist.

Property	Description
DataSourceDescription	The Description for the Data source. Usually the ODBC DSN or a File Name.
Name	The Name of the Service Adapter.
UserID	The UserID in which to use for the application. This is not the IONSID mentioned earlier.

Property	Description
Password	The password for the UserID supplied. Note, this password is only used when validating the settings during configuration. It is not stored.

4.2.2. Destination

The Destination Service Adapter's information. Many occurrences of this can exist.

Property	Description
DataSourceDescription	The Description for the Data source. Usually the ODBC DSN or a File Name.
Name	The Name of the Service Adapter.
UserID	The UserID in which to use for the application. This is not the IONSID mentioned earlier.
Password	The password for the UserID supplied. Note, this password is only used when validating the settings during configuration. It is not stored.

4.3. Message Body

The message body is also broken into two main sections. The Parameters Section and the Payload. It is in this section that the recipient of the message can get further information to process the request. It is also in this section where the Service Adapter will place the information or the data that was requested.

4.3.1. Parameters

This Section contains the parameters that are required for the message. Note that some messages do not require any parameters, in this case this section will be blank.

Property	Description
Name	The Name of the Parameter.
*Operation	The Operation to take with the Parameter. (ie. Equals, Not Equal, Greater Than....)

Property	Description
Value	The Value of the Parameter.

*Operation is not used at this time.

4.3.2. Payload

The Message Payload is the most complex section of the message. It contains the results of the message request. Three main sections make up the message payload. The PDXStatus Section, PDXLinks Section and the Payload Items section.

4.3.2.1. PDXStatus

This section contains information related to the completion of the message. If the message is successful, it will contain a status code and description indicating success. It is also here that you will find information about any errors that were encountered during the messages execution.

There can be many occurrences of this section.

Property	Description
StatusCode	The PDX Status code of the Error.
Description	A Description of the error.
OriginatedFrom	The Name of the DLL that the error occurred in.
ModuleName	The name of the Module that the error occurred in.
MethodName	The name of the Method where the error occurred.

4.3.2.2. PDXLinks

The PDXLinks section is only used when synchronizing information. It contains the Unique ID's for the Source and Target Application so that the Service Adapters know what to do when synchronizing, add or update. It also will has space so that the Service Adapter can tell PDX what it actually did with the data, add or update.

Property	Description
DataSourceDescription	The Description for a Data source. Usually the ODBC DSN or a File Name.

Property	Description
Name	The Name of the Service Adapter.
*SourceID	The Unique Identifier for the object in the Source Application.
*DestinationID	The Unique Identifier for the object in the Destination Application.
*PartyID	The Unique Identifier for the Object within PDX.
*ObjectType	Identifies the object as a Person or an Organization.
*ActionPerformed	The action that the Service Adapter performed on the Object. Add or Update.

* These properties can occur multiple times.

4.3.2.3. PayloadItem

The Payload Item section contains the results of the message. There can be one or more instances of a payload item. If the message was a search, then it would contain the search results. If the message was a getPerson the results would be for that request. As you can see it will vary depending upon the request.

Since a request can be sent to multiple destinations, we need to know what part, or Item, in the payload came from what Service Adapter. We accomplish this by including the following properties at the beginning of each payload item.

Property	Description
DataSourceDescription	The Description for a Data source. Usually the ODBC DSN or a File Name.
Name	The Name of the Service Adapter.

The rest of this section varies based on the request. For more information, please refer to Section 3 of this document. And also please refer to each Service Adapters documentation.

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

Application deficiencies found during scanning:

☐ Page(s)_____ of _____ were not present for scanning. (Document title)

☐ Page(s)_____ of _____ were not present for scanning. (Document title)

☐ ***Scanned copy is best available.*** some drawings are dark.